CMSC 733 - Project 4
Learning the Structure from Motion - An Unsupervised Approach

Nishad Kulkarni
A. James Clark, School of Engineering
University of Maryland, College Park
UID - 117555431, Email: nkulkar2@umd.edu

Saurabh Palande
A. James Clark, School of Engineering
University of Maryland, College Park
UID - 118133959, Email: spalande@umd.edu

I. Introduction

In this project, we learnt about estimating depth and pose (or ego-motion) from a sequence of images using unsupervised learning method implemented in the SfMLearner paper by David Lowe's team at Google. An unsupervised learning framework was presented for the task of monocular depth and camera motion estimation from unstructured video sequences. The next task was to improve the SfMLearner such that the error in pose and depth estimation is less than the original error.

II. SfM learner

The SfM learner is an end-to-end approach which allows the model to map directly from input pixels to an estimate of ego-motion (parameterized as 6-DoF transformation matrices) and the underlying scene structure (parameterized as per-pixel depth maps under a reference view).This method is unsupervised, and can be trained simply using sequences of images with no manual labeling or even camera motion information. Training examples to the model consist of short image sequences of scenes captured by a moving camera.

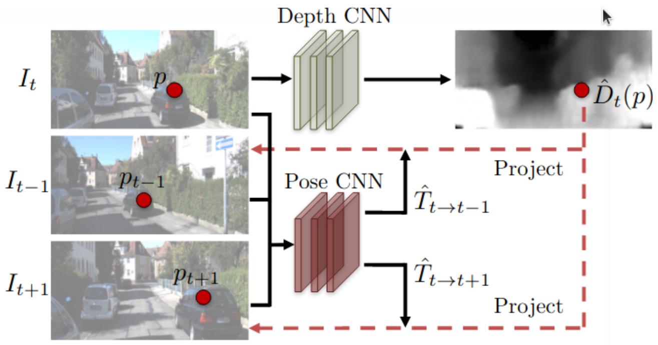

Fig. 1. Model Pipeline

Overview of the supervision pipeline based on view synthesis is shown in Figure 1. The depth network takes only the target view as input, and outputs a per - pixel depth map $\hat{D}_t$. The pose network takes both the target view (I_t) and the nearby/source views (e.g., I_{t-1} and I_{t+1}) as input, and outputs the relative camera poses. The outputs of both networks are then used to inverse warp the source views to reconstruct the target view, and the photometric reconstruction loss is used for training the CNNs. By utilizing view synthesis as supervision, the model is able to train the entire framework in an unsupervised manner from videos.

Assumptions used in the model:

1) The scene is static without moving objects.
2) There is no occlusion/disocclusion between the target view and the source views.
3) The surface is Lambertian so that the photo-consistency error is meaningful

Network Architecture

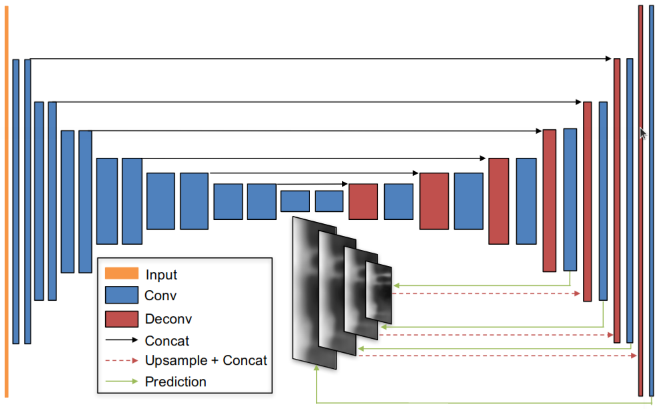

Fig. 2. Depth Network

Depth estimator: For single-view depth prediction, it adopts the DispNet architecture that is mainly based on an encoder-decoder design with skip connections and multi-scale side predictions. All conv layers are followed by ReLU activation except for the prediction layers. For the predicted layers they use

$$\frac{1}{\alpha\ sigmoid(x) + \beta} \tag{1}$$

with $\alpha = 10$ and β = 0.01 to constrain the predicted depth to be always positive within a reasonable range.

Pose Network The input to the pose estimation network is the target view concatenated with all the source views (along the color channels), and the outputs are the relative poses between the target view and each of the source views. The network consists of 7 stride-2 convolutions followed by a 1 * 1 convolution with 6 $*$ (N - 1) output channels (corresponding to 3 Euler angles and 3-D translation for each source view). Finally, global average pooling is applied to aggregate predictions at all spatial locations. All conv layers are followed by ReLU except for the last layer where no nonlinear activation is applied.

Explainability network The explainability prediction network shares the first five feature encoding layers with the pose network, followed by 5 deconvolution layers with multi-scale side predictions. All conv/deconv layers are followed by ReLU except for the prediction layers with no nonlinear activation. The number of output channels for each prediction layer is 2 * (N - 1), with every two channels normalized by softmax to obtain the explainability prediction for the corresponding source-target pair.

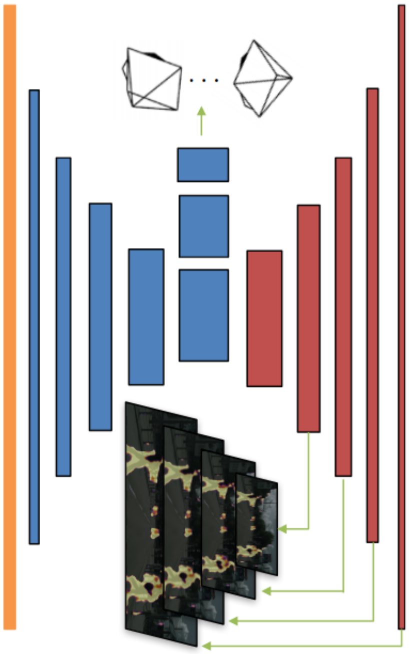

Fig. 3. Pose/Explainability Network

III. Network Modifications

The following sub-sections explain in detail the modifications done to improve the SfM learner.

A. Using Multiple views for Depth Estimation

The SfM-Learner only takes a single target view as input to compute depth map from the network and outputs a depth map. By using both target and source views as input it provides a sharper depth map image and the network also tends to converge with lesser iterations than the single view implementation. For every target image, we also input successive frames t + 1 and t - 1 as this eliminates noises induced in depth map due to various factors. The multiple views leverages the relationship between pixels over multiple views to calculate depth and hence provides a better depth map.

B. Using Structural Similarity loss

SfM-Learner uses photometric loss for training it's network. Using the photometric loss gives good results but it makes certain assumptions such as scenes need to have a constant brightness and luminosity. This constraint does not necessarily hold true in all cases and doesnot make the model robust. To solve this problem we added a different kind of metric Structural similarity Index (SSIM). The Luminance of surface of an object is a product of illumination and reflectance, but the structure of an object are independent of illumination. SSIM provides a robust metric for measuring perpetual differences between two images by considering the 3 factors of luminance, contrast and structure. We then added the SSIM loss term to the photometric loss. Since Structural similarity index is usually maximized (with the maximum value being 1), we minimize the below function

$$L_{ssim} = \sum_{s} \frac{1 - SSIM(I_t, I_s)}{2} \tag{2}$$

C. Using Adaptive Learning Rate

The SfMlearner implemented in the original paper used a constant learning rate of α = 0.0002. With a batch size of 4, this takes about 200 epochs before the network converges. Hence we implemented an adaptive learning rate, which starts with a high value of 0.002 but gradually reduces at every step of 10,000 iterations (15 epochs).

D. Changing data augmentation

SfM-Learner already includes some data augmentation by randomly scaling and cropping the input data. In addition to that, we added random shifting of the gamma values (making regions darker or lighter), randomly changing brightness and color of the images. These lead to decrease in loss.

IV. Results

The graph of our total loss is shown in Figure below:

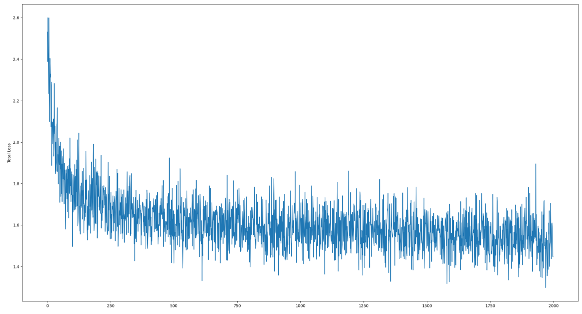

Fig. 4. Total Loss

We used the trained model to calculate the depth map of 5 images from the KITTI dataset - sequence 15. The depth map in the middle is the output from original SfM learner and the rightmost image is the depth map from our SfM learner. The results are shown below:

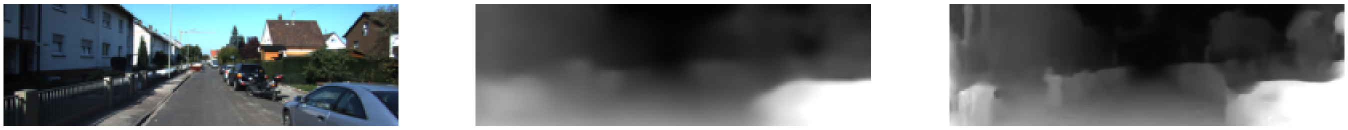

Fig. 5. Result - 1

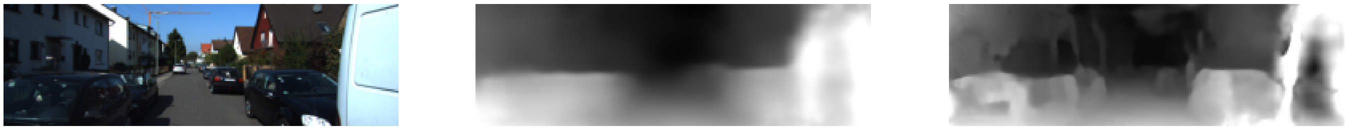

Fig. 6. Result - 2

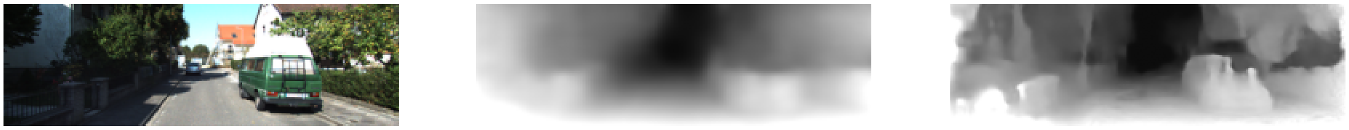

Fig. 7. Result - 3

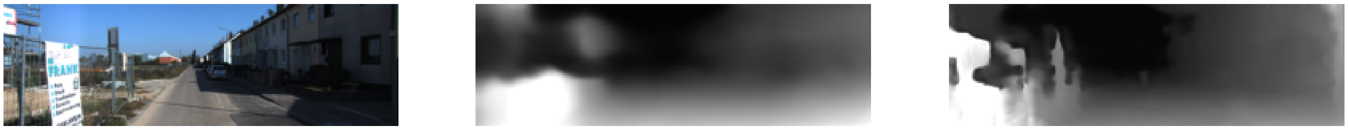

Fig. 8. Result - 4

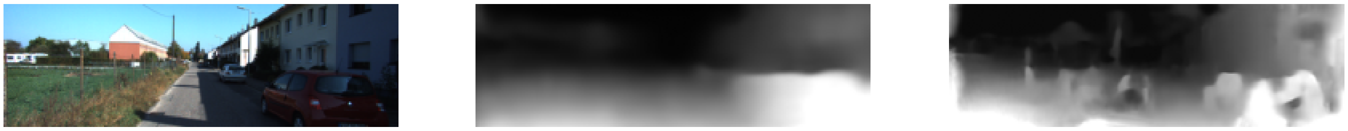

Fig. 9. Result - 5

References

[1] https://cmsc733.github.io/2022/proj/p4/

[2] https://arxiv.org/pdf/1704.07813.pdf

[3] https://github.com/tinghuiz/SfMLearner